

# RESET

```
RESET [INITIAL] operand1 ...
```

Operand	Possible Structure	Possible Formats	Referencing Permitted	Dynamic Definition
Operand1	S A G M	A N P I F B D T L C G O	yes	yes

## Function

The RESET statement is used to set the value of an operand(s) to a null value, or to an initial value as defined in a DEFINE DATA statement.

If *operand1* is a DYNAMIC variable, it will be reset to a null value with the length the variable currently has at the time the RESET statement is executed. The current length of a DYNAMIC variable can be ascertained by using the system variable \*LENGTH. For general information on DYNAMIC variables, see your Natural User's Guide.

(In reporting mode, the RESET statement may also be used to define a variable, provided that the program contains no DEFINE DATA LOCAL statement.)

## INITIAL

RESET (without INITIAL) sets the value of each specified field (*operand1*) to a null value.

RESET INITIAL sets each specified field to the initial value as defined for the field in the DEFINE DATA statement. If no initial value is defined for a field, it will be reset to a default initial value (see below).

If you apply RESET INITIAL to an array, it must be applied to the entire array (as defined in the DEFINE DATA statement); a RESET INITIAL of individual array occurrences is not possible.

RESET INITIAL of fields resulting from a redefinition is not possible either.

RESET INITIAL cannot be applied to database fields.

RESET INITIAL cannot be applied to DYNAMIC variables.

## Default Initial Values

If you specify no INIT or CONST value in the DEFINE DATA statement, a field will be initialised with a default initial value depending on its format.

## Example

```

/* EXAMPLE 'RSTEX1': RESET
/*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME (A10)
1 #BINARY (B4) INIT <1>
1 #INTEGER (I4) INIT <5>
1 #NUMERIC (N2) INIT <25>
END-DEFINE
/*****
LIMIT 1
READ EMPLOY-VIEW
/*****
WRITE NOTITLE 'VALUES BEFORE RESET STATEMENT:'
WRITE / '=' NAME '=' #BINARY '=' #INTEGER '=' #NUMERIC
/*****
RESET NAME #BINARY #INTEGER #NUMERIC
WRITE /// 'VALUES AFTER RESET STATEMENT:'
WRITE / '=' NAME '=' #BINARY '=' #INTEGER '=' #NUMERIC
/*****
RESET INITIAL #BINARY #INTEGER #NUMERIC
WRITE /// 'VALUES AFTER RESET INITIAL STATEMENT:'
WRITE / '=' NAME '=' #BINARY '=' #INTEGER '=' #NUMERIC
/*****
END-READ
END

```

```

VALUES BEFORE RESET STATEMENT:

NAME: MORENO      #BINARY: 00000001 #INTEGER:          5 #NUMERIC:  25

VALUES AFTER RESET STATEMENT:

NAME:             #BINARY: 00000000 #INTEGER:          0 #NUMERIC:   0

VALUES AFTER RESET INITIAL STATEMENT:

NAME:             #BINARY: 00000001 #INTEGER:          5 #NUMERIC:  25

```